

Tracking Truth Matching

- © Existing Track truth matching tools
- © New truth tool: TrkGTrackMatchByHits
- © Results using TrkGTrackMatchByHits

Existing Track Truth Tools

◎ **Micro association**

◎ **Based on track parameters at the origin**

- Insensitive by construction to decays-in-flight, etc

◎ **Recently improved by Amadeo Parazo**

- Fewer ghost/looper associations

◎ **TrkGTrkMatch (Luca Lista)**

◎ **Makes a bi-directional map between Reco tracks and GTrks**

- Uses hit->Gtrack mapping
- Association 'quality' is the number of associated hits

◎ **A framework module which puts data in the event**

- Part of TrkFinalSequence

◎ **Used by Emc,lfr,Dch**

- Typically only the 1st match (most hits) is used

◎ **Double-loop implementation ([gtracks[reco-tracks]])**

TrkGTrackMatchByHits

- ◎ **Maps reco tracks (and HOTS) onto gtracks**
 - ◎ Association by **normalized** hit weighting
 - Uses new class AstFracMap (AssocTools package)
 - Background hits affect the normalization
 - ◎ **Only active HOTS are used to define track association**
 - Can be computed separately for different mass hypos
 - ◎ **Uses digi->gtrack matching (for mixed-signal hits in Svt)**
 - ◎ **Also provides a HOT->Gtrack map**
 - Can look for split-association hits
 - Can look at HOT association by spatial, other parameters
 - Works for inactive hots as well
- ◎ **Utility class, not a framework module**
 - ◎ Mapping is done on construction
 - ◎ Can be put in the event (or not)
- ◎ **Single-loop implementation**

Results on 1K SP3 B_0B_0

- ◎ **Look at chisquared consistencies (probabilities) as a function of MC truth association for tracks and hits**
 - ◎ Association seems reasonable
- ◎ **Small probabilities are seen even for correctly-associated hits on ‘perfectly’ associated tracks**
 - ◎ Hard scattering?
 - ◎ Svt cluster splitting/merging?
 - ◎ Hit reconstruction errors?
 - ◎ Looper confusion?

Conclusions

- © **TrkGTrackMatchByHits** is a new tool for studying tracking using MC truth
- © **Initial results indicate DCH hit errors are sometimes underestimated**